A User-Oriented and Context-Aware Web Services Composition

Yiqiang Li
College of Software, Northeastern University, Shenyang, P.R.China
Email: yiqli@yahoo.com

Zhiliang Zhu
College of Software, Northeastern University, Shenyang, P.R.China
Email: zzl@mail.neu.edu.cn

Abstract—**The creation of value-added Composite web services provides an opportunity to B2B collaborations. The process of composing services is also a process of making services cooperative with each other in order to achieve the composing goal. This paper researches on how to compose Web Services in order to implement personalizing application. And a users-oriented composing approach is proposed in this paper which implements the virtualization of Web Services through 3-level mappings in order to map the user-familiar business services to physical Web services, implements the transformation from composite business service process to composite Web service process through planning techniques and through service selection makes the composite Web service meets users' requirements. The research shows that the prototype system of this approach is used well by normal business users and can also improve the effectiveness of formalization of Composite Web service.**

Index Terms—**Web Service Composition; Web Service; Virtualization; Web Service Selection; Business Service**

I. Introduction

With the development of Web services, people cannot be satisfied with current services. How to draw on available services to compose a new value added services gives us a lot of chances as well as challenges. The process of composing web services is a process where component services are identified and the dependencies between services are also described. For web services are in a high dynamic environment, therefore building composite services with an automated or semi-automated tool is critical. For web services are running in a context environment, composing services based on context is needed [1].

Workflow is touted for its intuitional ability that can express the whole composing process in an understandable way. And workflow composition normally can achieve the satisfied results through the participation of users, while such results can be ensured with a default assumption that users must understand the whole business process clearly and be familiar with the workflow modeling techniques. Thus, such assumption limits the range of workflow approach that only orients the professional designers. Although some efforts have been done for this limitation by employing templates for normal users through modifying the correct composing process by professional ones to achieve users' own requirement, such approaches always cannot work well for the templates may be not available for users' requirement. For such reason, workflow itself cannot perform well when doing the modeling task. Although AI technique claiming for its high inferential ability can form a composing process without the help of users, some defects must be addressed. The biggest one is that for the whole AI planning free of users' participation, the ultimate results are always uncertain.

Thus, after considering that end users are likely to express their own preferences in the modeling and comparing with the workflow and AI, this paper presents a new approach to model the composing process. Attraction for workflow's intuitional ability, firstly, users can use it to construct a rough model where they can only focus on what they want to do neither considering the control flow or data flow and we call this modeling process a user-defined functional modeling; and then the system-supported modeling process, based on the power of AI planner, uses it to reasons and transfers the rough one into a refined one; and finally in order to select the most qualified composite service, a process of QoS-driven selection is done which we call the instance-oriented modeling.

This paper is organized as follows: in section 2, some related works of composing web services is illustrated; and then, we discuss something that support the proposed approach in order to obtain a reasonable and qualified service composition; after that, features of the proposed approach is introduced where we discuss the proposed approach in detail; and next, through a prototype system, we give some evaluations towards our approaches; finally, we summarize this paper.

II. Related Works

Constructing application on the higher level is a traditional concept. Business Process Modeling Language (BPML) provides a business modeling approach which is based on workflow. UML as a tool is used to support the concrete physical process on the higher level [4]. Ref.[5]

doi:10.4304/jsw.7.4.878-883

researches on mapping the abstract workflow into physical resources scheduling workflow in the grid. And lots of works have been exerted on the web services composition [6,7,8]. Among these works, researches on user-oriented web service composition are less. Besides some workflow composition approach, AI planning techniques have been used in web service composition. Aiello provides a composition approach which is used planning techniques [9]. Ref. [10] proposes a rule based service composition system—SWORD. SWORD needs users to describe state and has no support for automatic discovery of services. Ref. [2] proposes an interface matching based web service composition approach which improve the matching precision with the help of domain ontology. Ref. [3] proposes a state calculus based web service composition approach which views services as action and considers the execution state of services.

Current researches on composition of web services as discussed above seldom exert effort on how to assist end users to compose services personalized. And some researches solve this problem only by adding semantic information throughout the composing. The approach proposed in this paper, which is an integration of workflow and AI planning through service virtualization, can enable end users to compose a correct and executable composite service.

III. Preliminaries

A. An overview of How to Form a Composite Web Services

After carefully analyze current system for service composition, we propose a system for composition (Figure 1), where service designers firstly use a service modeler to compose a service based on the approach we proposed as discussed in section 4; and such approach can be done through service virtualization; finally, the composite web services will be transported to a executive engine which will return the result of composite one.

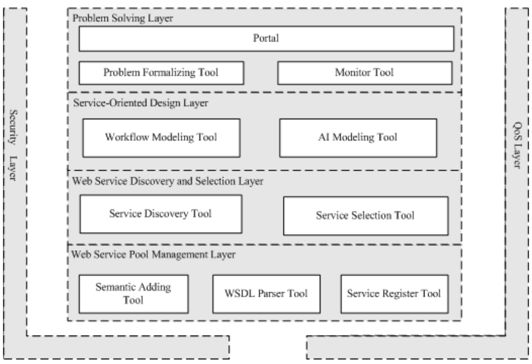

Figure 1. An overview of our system of composing web services

B. Service Virtualization

Service virtualization is used to map the business services into Web services in order to make the composition less complexity especially for the end users. For the end users, the Service Virtualization can make the Web services be understood and used by end users transparently. And we will introduce some basic concepts firstly and then shows the process of Service Virtualization.

Definition 1. Web Service (*WS*)

A Web service can be defined as a 7-vector: $WS = \langle S, F, I, O, Pre, Eff, Q \rangle$, where S is the basic description of *WS*, including text description and business information; F is the functional description in certain business domain; I is the set of input messages; O is the set of output messages; Pre is the precondition of execution; Eff is the effect of execution; Q is the quality information:

There exist equal relation and substitute relation between services.

- Equal Relation

If Web Service WS_1 and WS_2 satisfy (1), then there exists equal relation between them. We call WS_1 is equal to WS_2 and WS_2 is equal to WS_1 which can be expressed as $WS_1 = WS_2$.

$$\begin{aligned} &WS_2.F = WS_1.F \\ &WS_2.I = WS_1.I \\ &WS_2.O = WS_1.O \\ &WS_2.Pre \Leftrightarrow WS_1.Pre \\ &WS_2.Eff \Leftrightarrow WS_1.Eff \end{aligned} \tag{1}$$

- Substitute Relation

If Web Service WS_1 and WS_2 satisfy (2), then there exists substitute relation between them. We call WS_1 can be substituted by WS_2 which can be expressed as $WS_2 \rightarrow WS_1$.

$$\begin{aligned} &WS_2.F = WS_1.F \\ &WS_2.I \subseteq WS_1.I \\ &WS_2.O \supseteq WS_1.O \\ &WS_2.Pre \Rightarrow WS_1.Pre \\ &WS_2.Eff \Rightarrow WS_1.Eff \end{aligned} \tag{2}$$

If $WS_2 \rightarrow WS_1$ and $WS_1 \rightarrow WS_2$, then $WS_1 = WS_2$.

Based on equal relation between services, Web Services with equal function relations can be clustered together. And we can use cluster service to signify these services.

Definition 2. Cluster Service (*CS*)

A Cluster Service can be defined as a 5-vector: $CS = \langle F, I, O, Pre, Eff \rangle$, where F is the functional description in certain business domain; I is the set of input messages; O is the set of output messages; Pre is the precondition of execution; Eff is the effect of execution.

There exist substitute relation and equal relation between Cluster Services.

- Equal Relation

If Web Service CS_1 and CS_2 satisfy (3), then there exists equal relation between them. We call CS_1 is equal to CS_2 and CS_2 is equal to CS_1 which can be expressed as $CS_1 = CS_2$.

$$\begin{aligned} &CS_2.F = CS_1.F \\ &CS_2.I = CS_1.I \\ &CS_2.O = CS_1.O \end{aligned} \tag{3}$$

- Substitute Relation

If Cluster Service CS_1 and CS_2 satisfy (4), then there exists substitute relation between them. We call CS_1 can be substituted by CS_2 which can be expressed as $CS_2 \rightarrow CS_1$.

$$\begin{aligned} &CS_2.F = CS_1.F \\ &CS_2.I \subseteq CS_1.I \\ &CS_2.O \supseteq CS_1.O \end{aligned} \tag{4}$$

Based on the equal relation, Cluster Services with equal relation can be clustered together.

Definition 3. Business Service (*BS*)

A Business Service can be defined as a 3-vector: $BS = < F, I, O >$, where F is the functional description in certain business domain; I is the set of input messages; O is the set of output messages.

Based on definition 1, 2 and 3, we can form the mapping relationship between user-concerned business service and physical Web services to implement the service virtualization. Through doing this, the end users can just focus on their familiar business services while need not concern whether their composite process can be implemented by physical services. The mechanism of service virtualization can be showed in Figure 2.

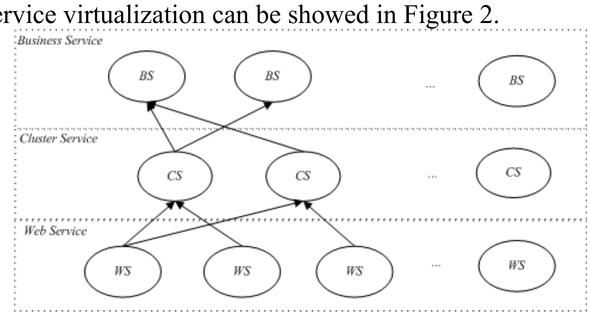

Figure 2. The mechanism of service virtualization

Mapping 1. Map Business Service into Cluster Service

For a Business Service *BS*, according to definition 2 and 3, it can be mapped into a set of Cluster Services.

$$[CS]_{BS} = \{CS | CS.F = BS.F \wedge CS.I \subseteq BS.I \wedge CS.O \supseteq BS.O\} \tag{5}$$

After adding executive conditions on $[CS]_{BS}$, *BS* can be mapped into a subset of $[CS]_{BS}$ as (6) shows.

$$\begin{aligned} [CS]_{BS|_{pre,eff}} = \{&CS \mid CS.F = BS.F \wedge CS.I \subseteq \\ &BS.I \wedge CS.O \supseteq BS.O \\ &\wedge CS.Pre \Rightarrow pre \wedge CS.Eff \Rightarrow eff\} \end{aligned} \tag{6}$$

Mapping 2. Map Cluster Service into Web Service

For a Cluster Service *CS*, according to definition 1 and 2, it can be mapped into a set of Web Services as follows:

$$\begin{aligned} [WS]_{CS} = \{&WS | WS.F = CS.F \wedge WS.I \subseteq CS.I \wedge WS.O \supseteq CS.O \wedge \\ &WS.Pre \Rightarrow CS.Pre \wedge WS.Eff \Rightarrow CS.Eff\} \end{aligned} \tag{7}$$

After adding non-functional condition on $[WS]_{CS}$, a *CS* with non-functional requirement *q* can be mapped it the subset of $[WS]_{CS}$ as follows:

$$\begin{aligned} [WS]_{CS|_q} = \{&WS | WS.F = CS.F \wedge WS.I \subseteq CS.I \wedge WS.O \supseteq CS.O \\ &\wedge WS.Pre \Rightarrow CS.Pre \wedge WS.Eff \Rightarrow CS.Eff \wedge WS.Q \Rightarrow q\} \end{aligned} \tag{8}$$

The algorithm for implementing the mechanism is illustrated as follows. According to the domain ontology, business services are firstly constructed manually by system manager. According to the relation between web services, the cluster services are constructed automated based on Algorithm 1. And according to the relation between cluster services and business services, the cluster set will be formed which can be seen in Algorithm 2.

Algorithm 1. Clustering Web Services.
Input: Web_service_set: WS_Set l
Output: cluster_service_Set
begin
for each $ws_i \in$ WS_Set
if $!\exists cs \wedge WS.F = CS.F \wedge WS.I \subseteq CS.I \wedge WS.O \supseteq CS.O \wedge WS.Pre \Rightarrow CS.Pre \wedge WS.Eff \Rightarrow CS.Eff$ then
begin
form *cs*;
insert *cs* into cluster_service_Set;
insert ws_i into $[ws]_{cs}$;
end
if $\exists cs$ then insert ws_i into $[ws]_{cs}$;
return cluster_service_Set;
end

Algorithm 2. Constructing Cluster Services Set.
Input: Business Service: BS, cluster_service_Set: CS_Set
Output: cluster_service_Set: $[CS]_{BS}$
begin
for each $cs_i \in$ CS_Set do
begin
if $cs_i.F \lhd BS.F$ then
insert cs_i into $[CS]_{BS}$;
end
return $[CS]_{BS}$;
end

The mechanism of service virtualization proposed in this paper can shield the complexity of composing task from end users who will just pay more attention to their familiar composite process of business services while need not concern how the business services will be substituted by physical services and how the process will be transformed. In the following, we will show our modeling approach which is an integration of workflow technique and AI planning.

IV. A New Approach of Modeling Composite Service Based on an Integration of Workflow and AI Planning

The modeling approach proposed here, is an integration of workflow and AI. Compared with traditional workflow approach, it enables the end users to compose services from the view of business while need not pay more attention to the difference between composite business service process and composite physical web service process. Meantime, it uses basic approaches of AI planning, which turns a rough composing process designed by users into a refined and correct one. Finally, based on the preference of users, the ultimate composite services can be drawn.

This approach divided the modeling process into 3 phrases (Figure 3):

- user-defined modeling phrase

  It is users to draw a composing process based on workflow technique where business services and preferences of users are defined;
- system-supported modeling phrase

  Based on the power of AI planner, this phrase uses it to reason and transfers the rough one into a refined one;
- instance-oriented modeling phrase

  Based on users' preferences, a QoS-driven selection is employed in order to form the ultimate composite process.

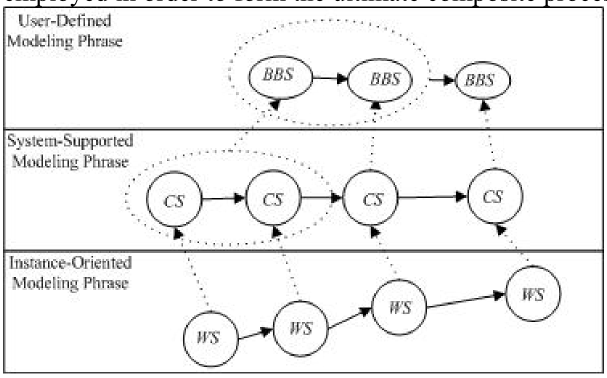

Figure 3. The proposed modeling approach

A. User-Defined Modeling Phrase

Composite service can be formed according to users' requirement. And in the user-defined modeling, a user graphic interface (Figure 4) is provided to users in order to help them draw a composite process which is based on basic idea of workflow.

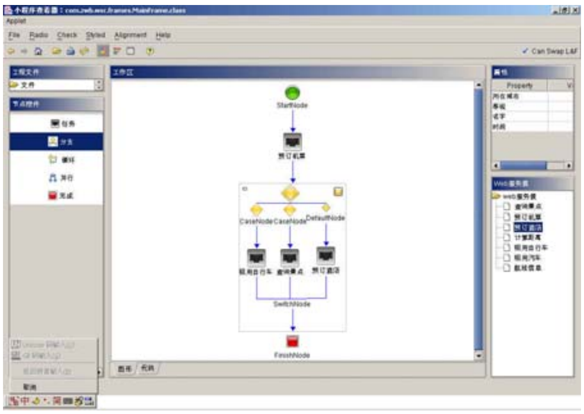

Figure 4. A GUI of proposed composing system

A composite process of "Travel Planner" which can be drawn by this GUI is shown in Figure 5. In Figure 5, the nodes are service clusters. However, such process is just a rough process where in order to execute the composite service in a predesired way, other services such as payment identification may be required and the dataflow of the process needs to identify clearly and correctly. That is to say, a refined process is needed through which the process will be corrected and refined.

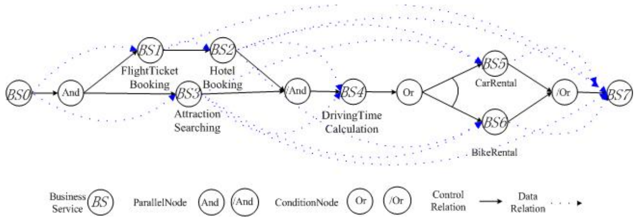

Figure 5. A composite process of "Travel Planner" expressed by statechart

B. System-Supported Modeling

In the user-defined process, users only draw a rough composing process where the granularity of composition is just in the level of business service. Thus, in this phase of modeling, the refined job will be done with the help of AI planning.

AI planning problem can be described as a 5-vector: $< S, S_0, G, A, \Gamma >$, where S is the set of possible state of the world; $S_0 \in S$ is the initial state of the world; $G \in S$ is the state of the goal; A is the set of actions which change the state of the world; $\Gamma \in S \times A \times S$ is a relation which defines the pre-condition and post-condition of an action. Here, A is the cluster service.

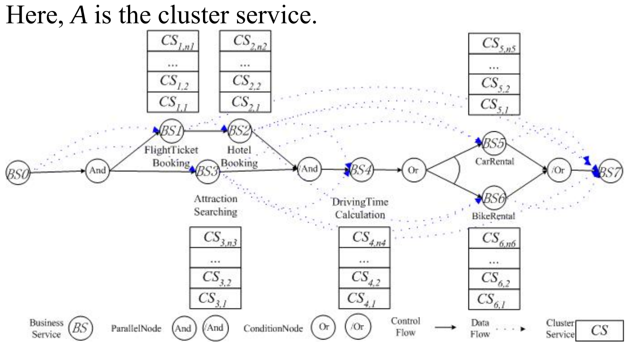

Figure 6. Mapping relation between business services and cluster services of Figure 5

The mapping relation between cluster services and business services is as Figure 6 shows and the refined process is through AI planning according to the pre-condition and post-condition of cluster services. Here, we use the technique of state searching to refine the process where forward planning is used to identify the inadequacy of some cluster services in the composite process, and based on these cluster services to do the backward planning, the algorithm of which is given as follows:

Algorithm 3. Planning

Input: Initial State, Goal

Output: Refined Process

1. put the initial state into the available set a[max]
2. for each business service of rough process do
3. if BS is the final one then
4. if Goal is satisfied with a[max] then
5. return refined one
6. else
7. rollback to BS i-1
8. if i-1 is –1 then
9. return fail
10. for each cluster services mapped into i do

11.if pre-condition of cluster service j can be satisfied by the item of a[max], then

12.put the effect of cluster service j into a[max], goto 2;

13.else, if pre-condition of cluster service j can not be satisfied by the item of a[max], then

14.goto step 15

15.if (exists a cluster service jk which provide effect as same as precondition of community j0) and (jk is not j0) then

16.if jk is satisfied with a[max], then

17.add cluster service jk before jk-1

18.else goto 8

19.else, rollback to the jk-1, goto 15

20.depose next cluster

Then, based on Algorithm 3, the ultimate refined process of Figure 5 is as Figure 7.

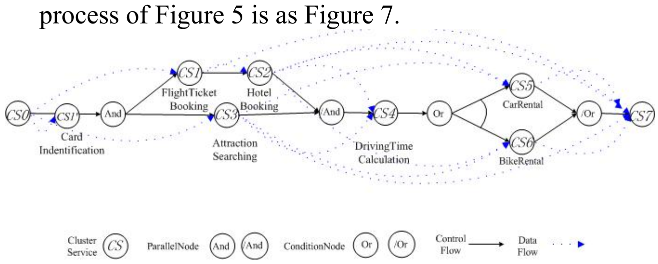

Figure 7. Refined composite Process of Figure 5

C. Instance-Oriented Modeling

Although Figure 7 is a refined and correct process and services which are mapped to certain cluster services can perform the ultimate goal of composition. In order to get a more qualified composite web services, an instance-oriented modeling is needed.

QoS is a broad concept which includes a range of non-functional properties, such as price, duration, availability and so on. *QoS* can be used to measure whether a web service can satisfy users' requirement and whether a web service can be executed correctly. For a web service *WS*, its *QoS* can be defined as a n-vector: $Q = < Q_1, Q_2, ... Q_n >$, where, Q_i is the non-functional property.

For Q_i, the value of it can be signified as $Q_i.Value$. And the weight vector assigned by users is $w = \left(w_k, 1 \le k \le n, \sum w_k = 1\right)$. Thus for a web service WS, its QoS Value can be calculated as: $WS.Q.Value = \sum w_k * WS.Q_k.Value$.

Then, for each cluster services in Figure 7, through mapping relation between cluster service and web service, the qualified services can be selected and the ultimate composite web service can be formed.

V. Implementation and Evaluation

In order to verify whether the approach proposed here is effective or not, we develop a prototype system SDOSWCP [11] the mainframe of which can be seen in Figure 4. End users can use this GUI to draw their composite process through dragging the nodes of business services. The service virtualization process is manually done by system manager as Figure 8 shows.

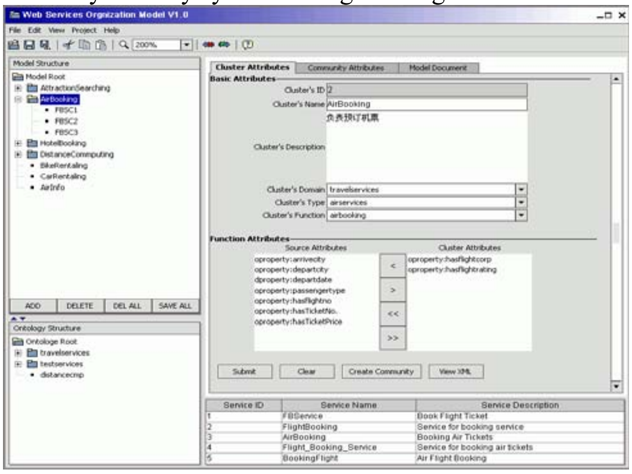

Figure 8. Mainframe of Service Virtualization

This system implements the basic idea of "User Oriented" concept addressed in this paper that users can compose services on the business level while need not concern how to form the composite web service process. Compared with other approaches, the composing approach proposed here have advantages:

First, workflow approaches is especially for professional designers who have a solid background of programming and familiar with WSDL and BPEL4WS. The approaches proposed here can be used by normal users which have a larger range of users.

Second, AI planning approaches is normally less satisfying because it is free of users' participation. The approaches proposed here used AI planning to do the refined process and the rough process is just designed by users thus it embodies the personalized requirements and more satisfying.

The aim of experimentation is to evaluate the proposed composition approach. The system-supported and instance-oriented selection is two key phrases which will affect the runtime performance. Thus, in the evaluation, this paper takes the sum of the execution time of this phrase as the runtime of the proposed approach.

We do the experiment on 6 situations where the number of Web services is 50, 100, 200, 400, 800, 1600 respectively. Give 15 composition request, compare the runtime of service composition among the approach presented in Ref.[2], Ref.[3] and ours. The experimentation result is shown in Figure 9.

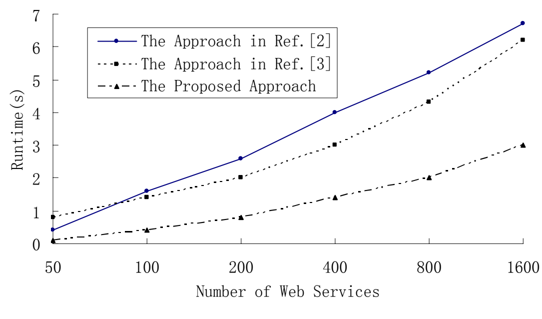

Figure 9. Comparison of Runtime

Figure 9 shows that with the increase of the number of the Web services, the runtime will be long. However, the proposed approach is always the most efficiency among the three approaches. This is because that the proposed approach uses the service mapping mechanism, which can minimize the searching space. Then, the efficiency will be improved.

VI. CONCLUSION

In summary, we have presented a new approach for Modeling Composite Web Services based on an integration of workflow and AI planning. Such approach we proposed, compared with workflow, lessen the user's participation and responsibility. On the other hand, compared with AI planning, enhance the compose possibility through the user-defined functional modeling.

Finally, we envision the following issues can be further explored to enhance performance composition of business process:

- The service virtualization will be done semi-automatically or automatically by support of ontology;
- The planning method will be improved in order to get the result more effectively.

REFERENCES

[1] L Baresi, D Bianchini, V D Antonellis, etl. Context-aware composition of E-Services. .

[2] A Lazcano, G Alonso, H Schuldt, C Schuler. The WISE approach to electronic commerce.Int J Comput Sys Sci Eng 15(5), 2000:343–355

[3] F Casati, S Ilnicki, L Jin, V Krishnamoorthy, M C Shan. Adaptive and dynamic service composition in eFlow.In: Proceedings of the CAiSE conference, Stockholm, June 2000,pp 13–31

[4] H Schuster, D Georgakopoulos, A Cichocki, D Baker. Modeling and composing service-based and reference processbased multi-enterprise processes.In: Proceedings of the CAiSE conference, Stockholm, June 2000, pp 247–263

[5] D Wu, E Sirin, J Hendler, D Nau, and B Parsia. Automatic web services composition using shop2. In Workshop on Planning for Web Services,Trento, Italy, June 2003.

[6] S A McIlraith, T C Son, H Zeng. SemanticWeb services.IEEE Intell Sys 16(2):46–53

[7] L Z Zeng and B Benatallah. QoS-Aware Middleware for Web Services Composition. IEEE Transactions on Software Engineering, Vol 30(5) May 2004 pp. 311-327

[8] Y B Han., S Tai and D Wikarski. A Dynamic Matching and Binding Mechanism for Business Services Integration. Proceedings of Engineering and Development of Cooperative Information Systems(EDCIS2002), September, Beijing, China, 2002, pp.17-20.

[9] J Cardoso., C Bussler, A Sheth and D Fensel. "Semantic Web Services and Processes: Semantic Composition and Quality of Service". On the Move to Meaningful Internet Computing and Ubiquitous Computer 2002, Irvine, CA, October 2002.

[10] A Patil, S Oundhakar and K Verna. "METEOR-s Web Service Annotation Framework". In Proceedings of International WWW Conference, New York, USA., May 2004.

[11] L Yang., Y Dai., B Zhang. A Service Design Oriented Web Services Composition Platform. Journal of Wuhan University. 2006, 11(1):160-164.

Yiqiang Li is bon in Shandong province, China, in 1969. He received the M.A in computer science from Northeastern University, China in 1997. He is a Ph.D candidate in computer application and technology in Northeastern University, China. His current research interests include cloud computing and service composition.

Zhiliang Zhu is born in Liaoning province, China, in 1962. He received the Ph.D. in computer science from Northeastern University, China in 2002. He is a professor at college of software, Northeastern University, China. His current research interests include cloud computing.