

<http://ansinet.com/itj>

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSI*net*

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Study of MDA Based Semantic Web Service Composition

¹Zhengdong Zhu, ²Yanping Chen, ¹Ronggui Lan and ¹Zengzhi Li

¹Department of Computer, Xi'an Jiaotong University, Xi'an 710049, China

²Department of Computer Science, Xi'an Institute of Post and Telecommunications, Xi'an 710121, China

Abstract: This study presents a Semantic Web Services composition method based on Model-Driven Architecture (MDA). It uses UML class and use-case diagram to build a model of static composition for OWL-S and it uses activity diagram to build a model of dynamic composition for OWL-S. Based on semantic matching degree, the matching methods are used to select a subset of the available web services and then the most suitable composite web services is identified. Through a translation to a verifiable Promela model, the composition UML models are verified by SPIN tool. The verified UML models are stored as templates in the knowledge base of ontology for reuse. Present method not only improves the development efficiency of the semantic web, but, also ensures the correctness of the composition process.

Key words: Semantic web service composition, model-driven architecture, semantic matching degree, SPIN tools

INTRODUCTION

The value of web services is the service reuse. Furthermore, the service composition is the sublimation of the service reuse. The key for effective service composition is the service modeling and the corresponding matching algorithm. Modeling web services composition is a complicated and error-prone process. If errors of composite web services are discovered after they have been put into service, there may be some serious effects and the cost to repair errors will be quite high. Therefore, in order to guarantee the correctness and the reliability of web services composition and verification for the process of Web services composition is necessary.

Currently, most of the semantic web service composition methods are dynamic composition based on the domain ontology (Lirong *et al.*, 2007; Jiangang *et al.*, 2005; Sayed *et al.*, 2008; Deke *et al.*, 2006), these methods mainly considered the quality of the composition, solved two heterogeneity problems of service composition and enhanced the accuracy of the service composition. However, these methods sacrifice the efficiency of the algorithm and do not use proper modeling tools, so their usability and practicality are directly affected. In addition, when selecting the most suitable composite web services, the selection rules based on functional property are IOPE semantic similarity match of examples of web service description and clients' requests (Martin *et al.*, 2006;

Lei and Horrocks, 2004; Constantinescu and Faltings, 2005), which do not consider influences of different semantic matching degree.

In the field of software, Model-Driven Architecture (MDA) is generally used to analyze, design and implement software systems. Now, modeling semantic web services composition with MDA is a hot research focus. Traditional model-driven construction methods of web service description can be broadly divided into two categories: one is based on the grammar of the WSDL or BPEL description, while the other utilizes semantic description of OWL or OWL-S. The former can deal with complex process but do not support the semantics, the latter deals with semantics but do not contain the composition process of complex control structure.

The Object Management Group (OMG) has defined the Unified Modeling Language (UML 2.0 Superstructure Specification, OMG adopted specification ptc/03-08-02. Technical Report, August 2003), In the use of UML modeling web services composition, Grønmo *et al.* (2005) gave the conversion rules between UML and OWL-S and activity diagram modeling are used for the semantic web services composition; Skogan *et al.* (2004) translated WSDL description of the web services into UML models, then composite web services described by UML to generate BPEL4WS and WorkSCo based on the grammar of XML; Timm and Gannod (2007) presented an improved model-driven approach, using UML class diagram and activity diagram to describe semantic web service

composition and using Ocl 2.0 specification (<http://www.omg.org/docs/ptc/05-06-06.pdf>, June 2005) to regulate the conditions on the control structure. However, these methods do not support semantic, that do not regulate all the control structures of the complex process and they did not verify the correctness of the process of web services composition.

In this study, an effective approach is presented to describe and composite semantic web services using UML. By using this method, an available and replaceable subset of web services can be selected. Then, the most suitable composite web service is identified and the correctness of this method is guaranteed according to the semantic matching degree defined by different weights based on the setting of the different matching degree.

WEB SERVICE COMPOSITE SYSTEM

The structure of the MDA based semantic web services composite structure is shown in Fig. 1. Web service provider provides the UML model in order to publish a web service and the UML model will be saved into the ontology knowledge base. The web services relationships ontology is set up using the existing web services. The relevant description of web services OWL-S is published to the Register Center.

Register Center stores web service registration information, such as UDDI, so that these Services can be found and invoked by any third parties.

Knowledge base is one part of the ontology and is also used to store web service UML models besides

proper functions of the ontology. For atomic web services, the generated model is the best web services abstract template found by the rules of definition. For composite web services, it is the web services generated through the composition that the users firstly made use of the web service, but it also can be customized, or is the abstract template of re-generated composition service through the matching selection method in the using process. Template is the service or composite services abstract model; there are no specific binding web services.

QoS (Quality of service) is a database system which includes all of the matrix definition of QoS, storage for each corresponding value of web services to support those who do not support the QoS property storage systems, is used with the regular match's selection of semantic web services.

Search engine can find the appropriate web services or composite web service template by using the matching method at knowledge base after user presents the request. It also can dynamically and adaptively find a subset of web service according to the semantic matching method with the template. Select fail means that none of the related information of all binding sub-services meets the requests according to the composition achieving template using the selection algorithm of usability-defined method based on functional and non-functional property (Zhengdong *et al.*, 2008).

When the composition web service template can not be found, composite engine automatically integrates the web service by using UML models and semantic matching

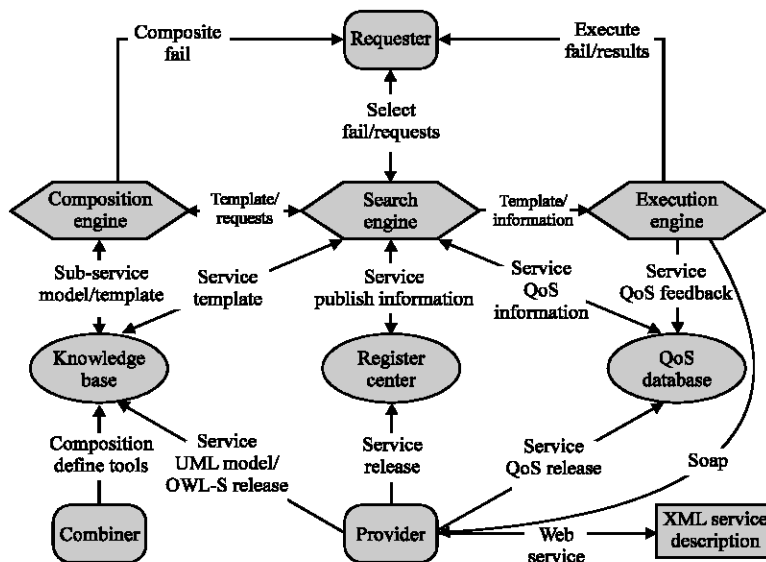


Fig. 1: Web services composition

composition approach based on the user's request. The obtained composition web service is verified by the use of the authentication method and then it is stored in knowledge base as a template. Composition engine can be integrated self-adaptive dynamically, can also be customized by the service combiner through definition interface.

Execution engine converts the template to specific work flow in BPEL4WS, manage and execute the web services providing by the service providers and achieve customer requests.

Additionally, in the whole process, the services matching method is used to match the request and the template, automatic service composition, sub-service binding template, respectively.

UML MODEL OF WEB SERVICE

The application of MDA to the analysis, design and implementation of suitable software systems has been recognized by the industry. Grønmo *et al.* (2005) have fully proved that MDA can meet the needs of semantic web service composition modeling in all the three aspects of UML, i.e., the expression, independence and readability of UML.

The UML model we proposed includes the static model and dynamic model. Static model is the description of the ontology and atomic services and the dynamic model is the description of the dynamic service composition. In the UML model, web service is described as an operation or behavior and there is an identified name and input-output in each operation or behavior. UML dynamic model defines strictly control flow and data streams between the operations.

Static model in UML: The ontology and atomic services are depicted through case diagrams. Moreover, the interaction between human and web service is modeled by use-case diagrams model.

Firstly, the hierarchy and relationship of ontology are modeled according to UML class diagrams: ontology can be generalized into domain ontology, while domain ontology can be generalized into a number of practical concepts. Then, web services are modeled in the same method. A web service is composed of one atomic service or several atomic services and each atomic service only has one function. Each operation in the class diagram is corresponded to a relevant atomic web service. At the same time, the input parameter of each operation is corresponding to the input of this atomic web service and the return value of each operation is corresponding to this atomic web service. Figure 2 is the class diagram of a web service FindCheaperService.

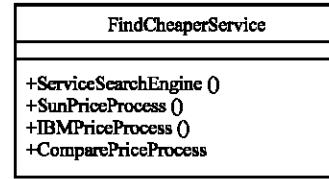


Fig. 2: Class diagram of FindCheaperService

Turn second to the model of interaction between human and web service. A use-case is corresponding to an atomic web service. A web service may have one atomic service or several atomic services need interact with actor including users. Modeling it with use-case diagram could show more clearly that which atomic service need to interact with an immovable actor and which participants would interact with an immovable atomic service.

Dynamic composition model in UML: The dynamic composition of web services is modeled by UML activity diagrams. Activity diagram is a technology to present process principle, business process and workflow. Because, the dynamic composition structure of OWL-S is composed of 8 basic structures in ServiceModel, the main work of modeling semantic web services composition with activity diagrams is how to describe the structures in ServiceModel using UML elements. These structures include sequence, split, split-join, choice, if-then-else, repeat-until, repeat-while and any-order.

Descriptions of these eight basic structures are disposed by activity diagrams, respectively. An action in an activity diagrams is corresponding to an atomic service. The most basic structures are sequence, split-join, choice, repeat-while and they are shown in Fig. 3. By the use of these structures, we may achieve the other four kinds of structures.

The sequence structure uses a straight-forward representation and it is a series of processes. These processes execute in a given order one after another, shown in Fig. 3a.

The split structure denotes concurrent execution of multiple processes and it ends if anyone of the processes is completely executed.

The split-join structure also represents concurrent execution of multiple processes, but it ends only when all of the processes are completely executed, shown in Fig. 3b.

The if-then-else structure has two paths of execution: the then path and the else path. Another version of the if-then-else structure is that the action in the then path or the else path maybe NULL action.

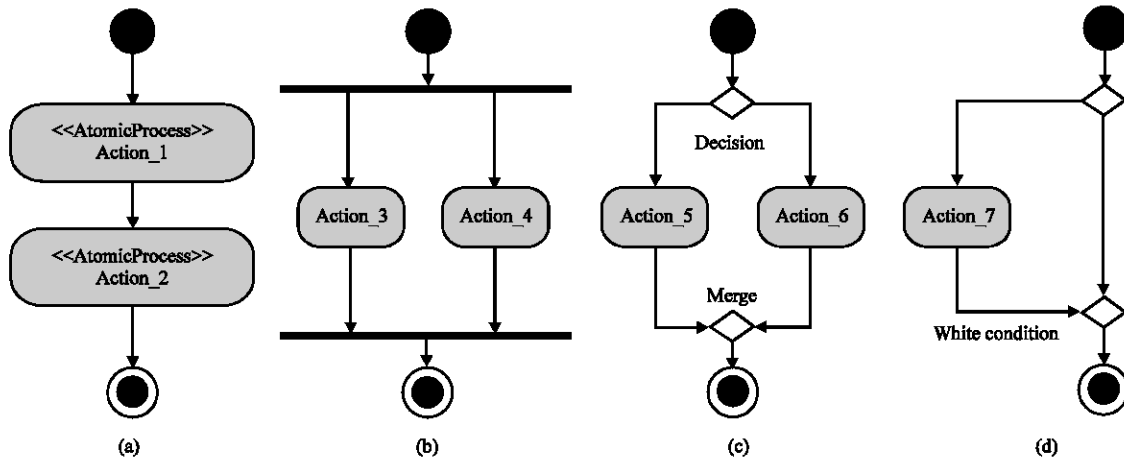


Fig. 3: UML models of the basic structures in OWL-S, (a) sequence, (b) split-join, (c) choice and (d) repeat-white

The choice structure is similar to the if-then-else structure. The difference is that the choice structure has two or more paths for chosen and only one path is selected to execute, shown in Fig. 3c.

There are two looping structures: Repeat-while and repeat-until. They are denoted by using analogous UML element merge and decision nodes. Repeat-while structure checks the condition first and then executes the process if the condition is true. While, the repeat-until structure executes the process until the condition is false, shown in Fig. 3d.

The any-order structure has two actions and both of them need to be executed once. However, they cannot be executed simultaneously. There are some paths in the picture all of which can be execute only once and the decision node chooses to execute one of the actions at the first time and chooses to exit at the second time.

In order to support all the information of OWL-S such as parameters, parameter's binding, condition and category etc., it's necessary to import tagged value, constraint, pin into present models. Pin in models is corresponding to the input/output in OWL-S and the parameters' binding is the relationship between pin and pin. Tagged value in the models is corresponding to the category and taxonomy and constraint in the models is corresponding to the conditions in OWL-S.

Tiwong and Kyongho (2007) gave the mapping relationship table between OWL and UML activity diagrams. In order to better illuminate how to model dynamic web services composition using activity diagrams. We present an activity model of composite web service FindCheaperService (Fig. 4).

When using the generating dynamic composition model in UML, there is some difference from dynamic model produced by modeling. Modeling UML dynamic model is only at the design stage and operation or

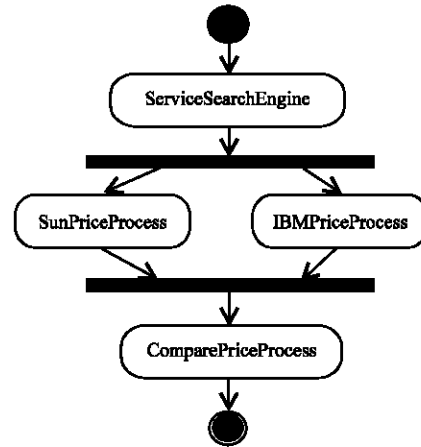


Fig. 4: UML activity diagram of FindCheaperService

behavior in the model is at an abstract level, while the using UML dynamic model is specific, utilizable.

WEB SERVICE MATCHING METHOD

In traditional component matching theory (Jorge and Amit, 2003; Rolland *et al.*, 1998), according to the service matching level (Deke *et al.*, 2006) and whether they are related to service behaviors and the required matching accuracy, the service matching can be divided into three levels: basic matching, signature matching and specification matching. In order to complete the complex web service composition, domain ontology and its reasoning are used at all the three levels of matching. Basic matching compares the name and description of the functional relevance; signature matching compares the input and output parameters of the name, type, range and direction of the correlation; specification matching compares the relevant of information.

Service matching is that the candidates from basic matching participate in the signature matching and then further participate in the specification matching. To complete the web services composition, in the entire course of the matching, domain ontology and its reasoning are used, basic matching here only compares the relevant of features description; this matching method based on semantic matching degree, introduced by the following section, are employed in signature matching and selection method based on usability of web service is used in specification matching (Zhengdong *et al.*, 2008),

It is assumed that the input number required is RI, the output number required is RO, the input number in advertisement is AI and the output number in advertisement is AO. When $AO \supseteq RO$, it means the service can meet the demand. If $RI \subset AI$, it's regarded there are no sufficient input parameters provided but can be supplied, it can still be thought meeting the demand. When $AI \subset RI$ or $AO \subset RO$, it means that it can't meet the demand. Considering comprehensively, compute through the following rules.

Definition 1: Matching degree of output: M.Output

Precondition: Search all the single web service meeting $AO \cap RO \neq \text{null}$ through output (referred to algorithm 1)

Algorithm 1 $AO \cap RO \neq \text{null}$

```
M.Output = 0
if  $AO \supseteq RO$  then
M.Output =  $(R_\alpha O/AO) \cdot \alpha_o + (R_\beta O/AO) \cdot \beta_o + (R_\gamma O/AO) \cdot \gamma_o$ 
else
M.Output = 0;
endif
```

$R_\alpha O$ means the output of precise matching required, $R_\beta O$ means output of replaceable matching required, $R_\gamma O$ means output of inclusive matching required, α_o , β_o , γ_o is the weight of precise, replaceable, inclusive matching of output, respectively

Definition 2: Matching degree of input: M.Input

Precondition: Multiple or single service meeting $AO \supseteq RO$ (referred to algorithm 2)

Algorithm 2 $AO \supseteq RO$

```
M.Input = 0
if  $AI \supseteq RI$  then
M.Input =  $(R_\alpha I/AI) \cdot \alpha_i + (R_\beta I/AI) \cdot \beta_i + (R_\gamma I/AI) \cdot \gamma_i$ 
else
M.Input = 0;
endif
```

$R_\alpha I$ means input of precise matching required, $R_\beta I$ means input of replaceable matching required, $R_\gamma I$ means input of inclusive matching required, α_i , β_i , γ_i , θ_i is the weight of precise, replaceable, inclusive, failing matching of input, respectively

Consequently, the semantic matching degree is $M.Degree = \min(M.Input, M.Output)$.

The services matching method is used to match demand and template, automatic service composition, sub-service binding template, respectively.

VERIFICATION OF SEMANTIC WEB SERVICE COMPOSITION MODEL

SPIN (Simple Promela Interpreter) is an assistant analysis and test tool for parallel systems and it is mainly used for formal verification of distributed systems. The input of SPIN is Promela and SPIN detects problems, tracks logic design errors and checks logical consistency by analyzing its Promela programs.

The main use of SPIN here is to describe the UML dynamic models i.e., the corresponding activity diagrams of the 8 basic structures in OWL-S by Promela language. Therefore, the generation of Promela programs from UML activity diagrams is expatiated.

The sequence structure in activity diagrams is equivalent to several processes executed in order in the Promela program. The Promela implementation of split-join is shown in Fig. 5 and the implement of the join synchronization is shown in lines from c to h. The implementation of a split statement would be identical, but lines 3 to 8 should be skipped.

The Promela implementation of the choice structure in activity diagrams is implemented using Promela's guarded non-deterministic choice statements $if :: fi$. Only one branch process which the guard condition is true can be executed according to the guard conditions and the other branch processes is skipped. The if-then-else structure is a special case of the choice structure, which only has two branch processes.

There are two kinds of loops in the UML activity models: repeat-while and repeat-until. Promela supports loops through the guarded $do :: od$ statements. The Promela implementation of repeat-while is as follows: the process enters the do loop and the condition whether c is true is checked during each iteration. If c is true, the process p is executed. Otherwise, the loop is broken. The

```
(a) pid childA = run A(childSync);
(b) pid childB = run B(childSync);
(c) if
(d) :: childSync??evel(childA), done->
(e) if
(f) :: childSync??evel(childB), done->
(g) fi
(h) fi
(i) SyncChan1_pid, done;
```

Fig. 5: Promela implementation of split-join

```

(a) Proctype FindCheaperService (chan syncChan, dataChan) {
(b)   Chan childSync = [2] of {int, mtype};
(c)   Chan childData = [1] of {int};
(d)   int service; int price;
(e)   pid childA = run ServiceSearchEngine (childSync);
(f)   if
(g)     :: childData?service -> childChan!service;
(h)     :: childSync?eval (childA), done;
(i)   fi
(j)   pid childB = run SunPriceProcess (childSync);
(k)   pid childC = run IBMPriceProcess (childSync);
(l)   if
(m)     :: childData? price -> childChan!price
(n)     :: childSync?eval (childB), done ->
(o)     if
(p)       :: childData?price -> childChan!price;
(q)       :: childSync??eval (childD), done;
(r)     fi
(s)   fi
(t)   pid child = run comparePriceProcess (childSync);
(u)   syncChan!_pid, done;
(v) }

```

Fig. 6: Promela implementation of the activity diagram of FindCheaperService

Promela implementation of repeat-until is analogous except two differences. The loop process p is executed once before checking the loop condition c and the condition c is a termination condition.

The any-order structure of activity diagrams consists of the if-then-else structure and the repeat-until structure, so the Promela implementation of any-order can be implemented through the Promela implementation of if-then-else and repeat-until. The process is described as follows: choose a branch to execute between process A and NULL first, next execute process B and then execute the branch which has not been executed before between process A and NULL, at last exit.

The UML model of FindCheaperService above is transformed into Promela program (Fig. 6). SPIN was adopted to verify its correctness.

The experiment shows that the UML model of FindCheaperService does not exist deadlock and satisfies the requirements of accessibility, safety and so on. This shows that our proposed approach for modeling semantic web services composition is correct. On the other hand, this also shows that our presented verification mechanism for semantic web services composition based on MDA is accurate.

The model test can validate two typical model properties, which are safety and liveness, by exhaustively enumerating possible execution process. The safety property includes correctness and security; the liveness property includes reachability and responsiveness. For instance, during the validation of the declaration to meet

correctness property, if the declaration is true in the SPIN validation of the Promela Norm, which means this declaration is also true in the original UML services assembling; if the declaration can not get through the SPIN validation, SPIN will present a counterexample to point out the condition in which the error happens. Simultaneously, the counterexample provided by SPIN can analyze and simulate the real process model. If the declaration with error lead to error behavior, which means a bona fide bug happened or else a spurious bug happened. With the same method, SPIN can also validate all the other properties. Consequently, the UML model can be verified by using the SPIN tools to validate the Promela model.

Finally, the standardized composite web services description will be generated from the verified UML composition model, such as OWL-S, BPEL4WS, etc. Then the generated composition web service description will be published into the web service registration center and QoS database using the same method as that with a single web service.

CONCLUSION AND FUTURE WORK

In internet applications, web services implement the reusability of the software. In this study, we use UML to describe the semantic web service description in OWL-S and we implement the MDA-based semantic web services composition. Present method can improve the development efficiency of semantic web services, the system usability and practicality. In MDA based composition, UML models are translated into verifiable description in Promela language and the SPIN tool is used to verify these descriptions and to ensure the correctness of the dynamic combination process. Abstractly describing the semantic web services at a higher level is also more favorable for dynamic binding of web services which is the foundation of composition web services. Besides, a matching method which defines different semantic similarity degrees and levels is proposed to improve the efficiency of the discovery and selection of web services. For semantic web service composition, verification and reusability, the model can meet customer requests at the greatest extent.

We will improve the target ontology further and expand the intelligent composition algorithm adaptability. The dynamic adaptive method will be adopted to obtain the weight value (Yang *et al.*, 2005). In addition, further study will be carried on web services verification methods, scheduling and implementation algorithms and reducing the computational complexity further.

ACKNOWLEDGMENT

This research is sponsored by the National Science Foundation of China NSFC under the Grant No. 60803162, 60673170 and 60773118.

REFERENCES

- Constantinescu, C.W.B. and B. Faltings, 2005. Flexible and efficient matchmaking and ranking in service directories. Proceedings of the IEEE International Conference on Web Services, ICWS 2005, Jul. 11-15, IEEE Computer Society, pp: 5-12.
- Deke, G., R. Yan and C. HongHui, 2006. A QoS-guaranteed and distributed model for web service discovery. Chinese J. Software, 17: 2324-2334.
- Grønmo, R., M.C. Jaeger and H. Hoff, 2005. Transformations Between UML and OWL-S. In: Model Driven Architecture-Foundations and Applications, Hartman, A. and D. Kreische (Eds.). Springer Berlin, Heidelberg, pp: 269-283.
- Jiangang, M., Z. Yanchun and L. Minglu, 2005. OMWSC-an ontology-based model for web service composition. Proceedings of the 5th International Conference on Quality Software, Sept. 19-20, IEEE Xplore, London, pp: 464-469.
- Jorge, C. and S. Amit, 2003. Semantic e-Workflow composition. J. Intell. Inform. Syst., 21: 191-225.
- Lei, L. and I. Horrocks, 2004. A software framework for matchmaking based on semantic web technology. J. Elect. Commerce, 8: 39-60.
- Lirong, Q., S. Zhongzhi and L. Fen, 2007. Agent-based automatic composition of semantic web services. Chinese J. Comput. Res. Dev., 44: 643-650.
- Martin, D., M. Burstein and J. Hobbs, 2006. OWL-S: Semantic markup for web services. <http://eprints.ecs.soton.ac.uk/12687/>.
- Rolland, C., C. Souveyet and A.C. Ben, 1998. Guiding goal modeling using scenarios. IEEE Trans. Software Eng., 24: 1055-1071.
- Sayed, G.H.T., M.N.W.K. Wan and I. Suhaimi, 2008. Semantic web service discovery and composition based on AI planning and web service modeling ontology. Proceedings of the 2008 IEEE Asia-Pacific Services Computing Conference, Dec. 09-12, IEEE Xplore, London, pp: 397-403.
- Skogan, D., R. Grønmo and I. Solheim, 2004. Web service composition in UML. Proceedings of the 8th IEEE International Enterprise Distributed Object Computing Conference, Sept. 20-24, IEEE Xplore, London, pp: 47-57.
- Timm, J.E.T. and G.C. Gannod, 2007. Specifying semantic web service compositions using UML and OCL. Proceedings of the IEEE International Conference on Web Services, Jul. 9-13, IEEE Xplore, London, pp: 521-528.
- Tiwong, K. and L. Kyongho, 2007. Describing semantic web service: From UML to OWL-S. Proceeding of the IEEE International Conference on Web Service, Jul. 9-13, IEEE Xplore, London, pp: 529-536.
- Yang, W.J., J.Z. Li and K.H. Wang, 2005. Domain-adaptive service evaluation model. Chinese J. Comput., 28: 514-523.
- Zhengdong, Z., M. Xianglin and H. Yahong, 2008. Usability-based general semantic web services discovery mechanism. J. Xi'an Jiaotong Univ., 42: 659-663.