2008

Adaptive task allocation for P2P-based e-services composition

Jun Shen
University of Wollongong, jshen@uow.edu.au

Shuai Yuan
University of Wollongong, sy242@uow.edu.au

Publication Details

Shen, J. & Yuan, S. (2008). Adaptive task allocation for P2P-based e-services composition. In M. Al-Akaidi, L. Cui, B. Hu, B. Hu, Z. Lin & Y. Zhang (Eds.), International Conference on Pervasive Computing and Applications (pp. 313-318). Piscataway, NJ: IEEE Conference Publications.

Adaptive task allocation for P2P-based e-services composition

Abstract

To effectively manage the task allocation, especially when handling with numerous different peers' qualities, is one of the greatest challenges to be faced in order to guarantee the success of P2P-based e-services composition. In this context, various QoS descriptive frameworks and Web services technologies (such as WSDL and BPEL) are being considered as the most affordable solutions to promote the performance of decentralized e-services, through applying strategies like QoS ontologies and related optimization algorithms globally or locally. Nonetheless, most P2P-based service selection and composition approaches applied nowadays lack dynamism and autonomy. In this paper, we first propose an extension of non-functional properties in WSMO, so that to globally facilitate dynamism and autonomous coordination in service compositions. Furthermore, taking into account a model driven approach, we design a planning algorithm to intelligently assign composition tasks to the most appropriate peers for different steps in a whole process. This algorithm is implemented in our prototype UOW-SWS via considering a typical LoanApproval scenario.

Keywords

Adaptive, task, allocation, for, P2P, based, services, composition

Disciplines

Physical Sciences and Mathematics

Publication Details

Shen, J. & Yuan, S. (2008). Adaptive task allocation for P2P-based e-services composition. In M. Al-Akaidi, L. Cui, B. Hu, B. Hu, Z. Lin & Y. Zhang (Eds.), International Conference on Pervasive Computing and Applications (pp. 313-318). Piscataway, NJ: IEEE Conference Publications.

Adaptive Task Allocation for P2P-Based e-Services Composition

Jun Shen, Shuai Yuan
Centre for Information Systems and Technology Research
School of Information Systems and Technology
Faculty of Informatics, University of Wollongong, Australia
Wollongong 2522, Australia
jshen@uow.edu.au, shuai.yuan@hotmail.com

Abstract

To effectively manage the task allocation, especially when handling with numerous different peers' qualities, is one of the greatest challenges to be faced in order to guarantee the success of P2P-based e-services composition. In this context, various QoS descriptive frameworks and Web services technologies (such as WSDL and BPEL) are being considered as the most affordable solutions to promote the performance of decentralized e-services, through applying strategies like QoS ontologies and related optimization algorithms globally or locally. Nonetheless, most P2P-based service selection and composition approaches applied nowadays lack dynamism and autonomy. In this paper, we first propose an extension of non-functional properties in WSMO, so that to globally facilitate dynamism and autonomous coordination in service compositions. Furthermore, taking into account a model driven approach, we design a planning algorithm to intelligently assign composition tasks to the most appropriate peers for different steps in a whole process. This algorithm is implemented in our prototype UOW-SWS via considering a typical LoanApproval scenario.

Keywords: Peer to Peer, Quality of Service, WSMO, Web services, Ontology

1. Introduction

Task allocation is an important issue in dynamic decentralised e-service application. Some existing approaches are based on Quality of Service (QoS) optimization. Menascé [7] defines QoS as "a combination of several qualities or properties of a service". It is a set of non-functional attributes that may influence the quality of the service provided by a resource and consequently represent key components of a Web Service Agreement [2]. In fact, non-functional features of Web services play a very important role in performance management of a composite Web service, and even spatial characteristics [13] also become a concerning aspect in decentralised service network. Recently, the increasing effort has been focused on how to describe and utilise those non-functional information to schedule an efficient services composition, especially in P2P-based or agents-based information systems.

It is obvious that the distributiveness, dynamics and heterogeneity of services become extremely important to both service requestors and service providers. Nevertheless, most research works presented so far are mainly syntactic and have not fully incorporated ontology model for service description and composition within real circumstances. Still, the selection and integration of a new service in an existing infrastructure is not automatic and requires a lot of human effort. Even though quite a few groups proposed numerous QoS specifications, most of them are extremely difficult to clarify the correlation between one another consistently.

Preferably, some non-functional properties in Web Service Modelling Ontology (WSMO) [10] can be employed as a discriminator factor to refine P2P-based Web services so as to facilitate a more effective schedule in business workflows. In this paper, we present an autonomous and scalable ontology-based methodology to describe QoS features of Web services in a P2P-based environment. Moreover, based on our ontology model, we design and implement an algorithm to plan the whole composition process and assign the tasks to the most appropriate peers in order to foster a better service composition.

The paper is structured as follows. Section 2 will explain basic knowledge of WSMO and a typical use case. Section 3 will introduce the design steps of WSMO extension, with a focus on modelling support for QoS characteristics. Beyond this, our generic algorithm for the task allocation process, which is based on unified correlation of different quality metrics, is also introduced. Section 4 presents implementation of the UOW-SWS prototype for a typical composition case. Section 5 will discuss the related research work in QoS descriptive framework and Web service selection methods. After that, our conclusions will be addressed with future work in Section 6.

2. Background and Motivation

In general, WSMO aims to create an ontology which can semantically describe a variety of perspectives of Web services, so as to solve the integration problem. Essentially, WSMO defines four high-level notions which relate to semantic Web services, namely Ontologies, Goals, Mediators and Web services [14].

Non-functional properties are usually utilised to describe non-functional aspects such as the creator and the creation date, and to provide natural-language descriptions, etc. All of the four WSMO elements [14] have their own non-functional properties. In this paper, however, our QoS extension is of the same nature as the notion of non-functional properties in "Web services". In other words, we mainly focus on the consideration of QoS, such as performance, availability, cost of distributed services, etc. The incorporated QoS properties could also be used in parallel with existing non-functional attributes proposed by other WSMO elements. Thus, it is consistent to consider QoS parameters as more general non-functional properties.

We develop the non-functional properties in WSMO in order to support adaptive P2P-based service composition. For example, Response Time, Availability, Reliability, Accessibility, etc. are very typical and necessary to describe a service provider's quality in a dynamic decentralised network. More importantly, geographic features [13] can be applied in these non-functional properties, as location information of peers is always needed as extremely useful and essential aspects to enhance P2P-based computing.

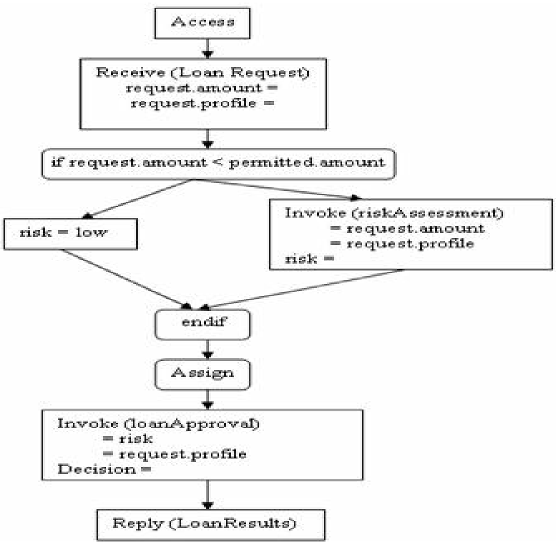

Figure 1: CFG in LoanApproval example

Our prototype, UOW-SWS, is a JXTA-based [4] peer-to-peer workflow information system upgraded from SwinDeW-B [16], which was designed and developed to overcome the problems like poor performance, poor scalability, unsatisfactory system openness, and lack of support for incomplete process. UOW-SWS have been functionally extended to incorporate WSMO features so as to facilitate Web services selection via QoS and spatial information.

There is a typical case, LoanApproval, used by many e-commerce application prototypes, so we'd like to utilise it in UOW-SWS to testify and demonstrate our selection method in an empirical way. Figure 1 shows the CFG (Control Flow Graph) [17] which depicts a typical loan application process. For the whole process, it consists of two small single services (i.e., task or activity): 'riskAssessment' and 'loanApproval'. 'riskAssessment' is to provide the service about evaluating customer's reputation and loan amount, so that it will generate the risk assessment of loan. Only when the risk assessment meets the requirement (e.g. higher reputation with more permitted loan amount) of 'loanApproval', can the loan application be approved; otherwise, the loan request will be rejected.

Technically, a coordinator in our prototype works similarly to the mediator in WSMO, which can adaptively organise the peer/agent selection process and distribute tasks subsequently. In this way, the decentralised run-time environment can be coordinated and self-managed effectively with services being located to wide area peer/agent hosts, who are able to communicate with each other according to a real business process agreement or standard workflow definitions. In this paper, we design an effective and qualitative way in P2P information systems to globally plan a composition process for the requested service.

3. Planning for Peers Composition

3.1 A generic peer selection method

In order to evaluate different non-functional properties of e-service peers, there are three important concepts in our design: *PreferedValueType, Weight,* and *Unified Value. PreferedValueType* has two kinds of values: "low" and "high". We utilise them to quantitatively identify two different types of properties among numerous non-functional properties in real use cases. For example, "ResponseTime" usually is expected as short as possible when choosing an appropriate peer, so the PreferedValueType of "ResponseTime" is "low". Likewise, "Distance" also usually relates to "low", as no one would choose a service with a long distance. However, "Reputation" and "AvailableDuration" often fit into "high", since their values are often expected as high as possible. Accordingly, all peers' various properties are viable to be categorised into the two types. With regard to "*Weight*", it indicates the importance and priority of certain properties during the service composition, so weight value varies from service to service, and from property to property. Lastly, "*Unified Value"* indicates the each peer's overall quality with numerically indicating results. With a set of equations as defined below, we can calculate a *"Unified Value"* so as to evaluate and rank each peer's overall capability to meet requirements against a requested service.

If "*PreferedValueType"* = "high", then the property ratio (PR) of a peer's service should be calculated by:

$$\mathrm{PR(i,j)} = \frac{\mathrm{nf(i,j)\text{ - }nf(min)}}{\mathrm{nf(max)\text{ - }nf(min)}} \tag{1}$$

"PR(i,j)" presents the ratio value of non-functional Property(j) of Peer(i), and *"nf"* stands for non-functional. nf(min) and nf (max) refer to the minimum and maximum value of the Property(j) among all relevant peers. On the contrary, if "*PreferedValueType"* = "low", then the ratio should be determined according to:

$$\mathrm{PR(i,j)} = \frac{\mathrm{nf(max)\text{ - }nf(i,j)}}{\mathrm{nf(max)\text{ - }nf(min)}} \tag{2}$$

Our main aim is to scale the value ranges with the maximum and minimum values by this means. Hence, any value with different "*PreferedValueType*" can be converted into the standardised value between 0 and 1. Through this approach, every property of each peer can be compared and evaluated fairly and also quickly.

Subsequently, all candidate peers' non-functional properties would be put in a matrix, looks like (for n properties in m peers):

$$Mnf = \begin{bmatrix} PR(1,1) & PR(1,2) & PR(1,3) & \ldots & PR(1,n) \\ PR(2,1) & PR(2,2) & PR(2,3) & \ldots & PR(2,n) \\ PR(3,1) & PR(3,2) & PR(3,3) & \ldots & PR(3,n) \\ \ldots & \ldots & \ldots & \ldots & \ldots \\ PR(m,1) & PR(m,2) & PR(m,3) & \ldots & PR(m,n) \end{bmatrix}$$

"Mnf" refers to matrix of non-functional properties. For uniformity, matrix *Mnf* has to be normalised to map all real values to a relatively small range through equations (1) (2), i.e., all elements of the final matrix are real numbers in the closed interval [0, 1].

Having *Weight (W)* values assigned to each property, we apply the following equation to generate the "*Unified Values (UV)*" for each peer:

$$UV = Mnf \times W,$$
$$i.e., UV(i) = \sum_{j=1}^{n} (PR(i,j) \times w(j)), i = 1..m \tag{3}$$

w(j) stands for a weight value of different property (j^{th}) for service composition. As a result, it is reasonable to indicate which peer (i^{th}) would be able to conduct a specific task more effectively, by means of achieving the highest value *UV(i), i* ranges from *1 to m.*

3.2 QoS features extended in WSMO

Based on [15], we define an extensible class QoSProperty which extends nonFunctionalProperties class in WSMO for P2P-based service selection.

Class nonFunctionalProperties
    ...other existing properties...
    hasQoSProperty **type** QoSProperty

Class QoSProperty **sub-Class**
  nonFunctionalProperties
    hasPropertyName **type** string
    hasPropertyValue **type** {int, float, long, others}
    hasPreferedValueType **type** {low, high}
    hasWeight **type** float

Each QoS Property is generally described by *PropertyName* and *PropertyValue*. For the purpose of QoS-based selection, there are two additional attributes defined, namely: *hasPreferedValueType* and *hasWeight.* The *hasPreferedValueType* is an object property representing the expected tendency of the value for the ideal attribute. The *hasWeight* is a value denoting the weight of the property, especially when synthetically measuring several different property metrics. In this context we define the weight value within range [0, 1], while different end users may have different weight values for their service requirements.

For instance, a peer's "ResponseTime" can be described in Web service profiles as following:

dc _"http://purl.org/dc/elements/1.1#",
webService _http://example.org/ LoanApprove
nonFunctionalProperties
  dc#title **hasValue** "Peer 1"
  dc#description **hasValue** "ResponseTime for LoanApprove process by peer 1"
  ……
  hasPropertyName **hasValue** _string ("ResponseTime")
  hasPropertyValue **hasValue** _int ("700")
  hasPreferedValueType **hasValue** _string ("low")
  hasWeight **hasValue** _float ("0.75")
endNonFunctionalProperties

3.3 Global planning algorithm

For the decentralised e-service application, each functional peer plays an important role in a composition process. Based on each peer's quality and performance, peer can contribute differently towards the whole service process. Importantly, since peers are usually associated with different service qualities and a Web service might be conducted by more than one functional peer. It is extremely necessary to find the best path with the most appropriate peers for the composition process. By this means, finding the executable and efficient path through those peers would definitely save much time and costs.

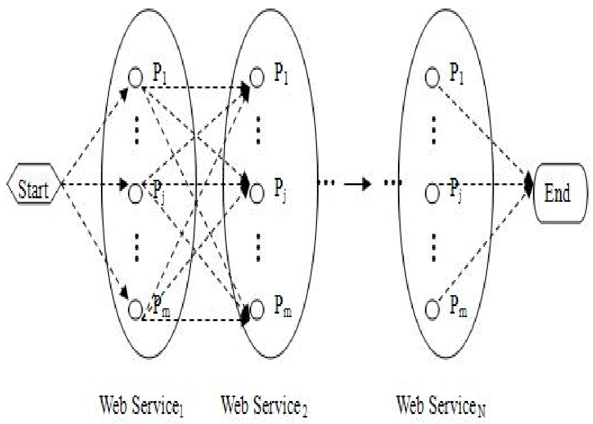

Figure 2: Peers Composition Process Model

The figure 2 describes the composition process in which there are many potential paths amongst peers, and we need to distinguish those peers from the same functions and reasonably plan a path for the whole service composition. To enable peers' coordinator intelligently select peers and plan a whole composition process, we design an algorithm to plan the best path through the service composition. The following is the pseudo code:

Begin Function Planning Peers (P_1, P_2, ... P_m) for Composition
  Initialise N= the number of Web services/tasks in required composition;
  for i=1 to N do
   for j=1 to m do
    getQoS(P_j) for Task$_i$;
    normalise input (P_j) using equation (1)/(2) in section 3.1;
    then store the normalised value into array (*Mnf*);
   end
   getWeight() for the different properties;
   calculate the unified values by using equation (3) in section 3.1;
   choose P_j which is with maximum unified value;
   assign Task$_i$ to P_j;
  end
end function

This algorithm aims to address the planning method with multiple peer profile specifications, and facilitate the above modelling approach. With regard to the loan use case, it is usually required that a selected peer should have better overall quality than others, thus a coordinator can apply this algorithm to efficiently allocate tasks to the most appropriate peers for the whole service composition.

4. Experiment

In our experiment, firstly, we assign two sets of random data (Figure 3) for two tasks ('riskAssessment' and 'loanApproval'), and then demonstrate the evaluation of four peers who are available in our loan case in UOW-SWS prototype, i.e., pre-deployed in the JXTA network.

For the whole composition, to effectively plan peers' tasks with combined QoS specifications is often a quite complex process, due to the diversity of various metrics with different value types, value range, and measurements. For example, figure 3 presents the complex quality status of candidate peers.

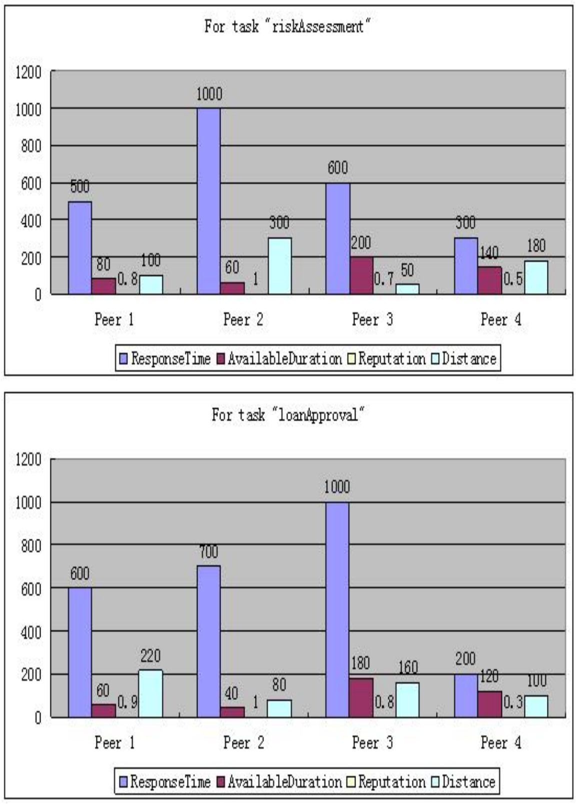

Figure 3: Peers' Qualities for Different Tasks

Taking account of correlations between those different specifications, we simplify and unify those combined various specifications so as to make the planning process less complicated and more effective. Regarding *Weight* we introduced in section 3.1, their values often vary from different requirements and situations in real environment. Based on the importance of properties, the weight value for the four properties can be W= $(0.8, 0.5, 0.85, 0.6)^T$ in the loan case, distinguishing ResponseTime, AvailableDuration, Reputation and Distance respectively.

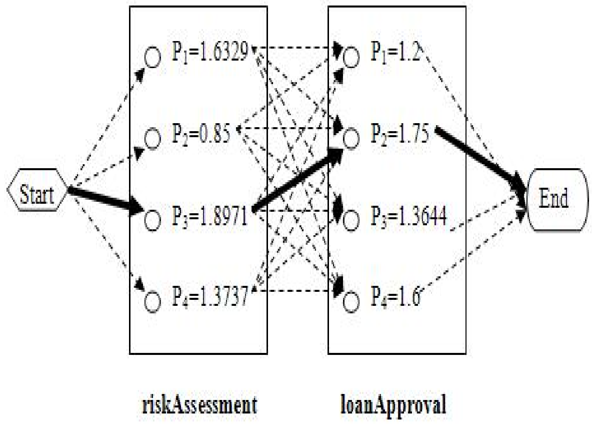

Figure 4: Results for planing the best path

In figure 4, for “riskAssessment”, the peers’ unified quality values are calculated in UV = $(1.6329, 0.85, 1.8971, 1.3737)^T$ by applying formulas in section 3.1. Likewise, for “loanApproval”, the four peers’ unified quality values are in $UV = (1.2, 1.75, 1.3644, 1.6)^T$. Accordingly, we select the peer who has the highest value for a task, so the best path is highlighted in the figure for the service composition.

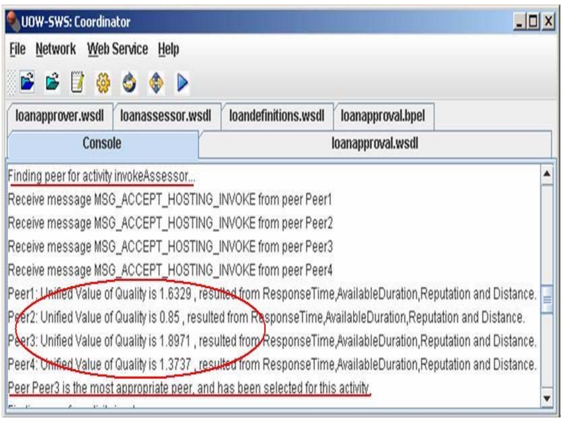

Figure 5: Assigning “riskAssessment”

We implemented the proposed algorithm from section 3.3 in our prototype. In the screenshots of UOW-SWS (Figure 5 and Figure 6), we can see the Coordinator peer precisely selected Peer 3 as the most appropriate one (for invoking the assessor) after a round of communications among peers. Figure 6 shows Peer 2 has been selected as the current best one as the service conductor for invoking the approver. This selection method for peers’ combined specifications is reasonably suitable and effective to be fully adapted to the real dynamic environment, especially in the sense of an autonomous way to effectively coordinate the composition process for decentralised service application.

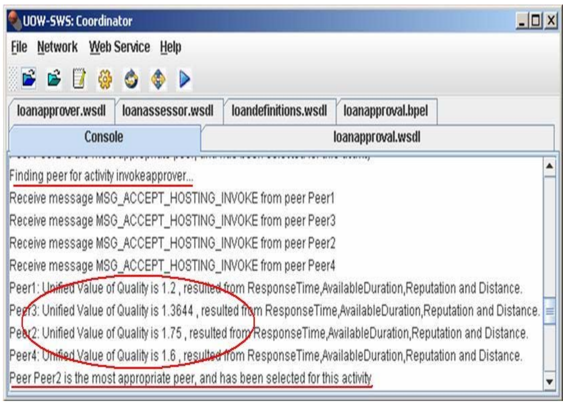

Figure 6: Assigning “loanApproval”

5. Related Work

Research work in the area of Web service QoS typically involves syntactic aspects. Most related works regarding the QoS in Web services focus on the development of QoS ontology languages and vocabularies, as well as the identification of various QoS metrics and their measurements with respect to semantic e-services. For example, [9] and [5] emphasized a definition of QoS aspects and metrics. In [9], all of the possible quality requirements were introduced and divided into several categories, including runtime-related, transaction support related, configuration management and cost related, and security-related QoS. Both of them shortly present their definitions and possible determinants. Unfortunately, they failed to present a practical methodology for real applications. In [8] and [15], authors focused on the creation of QoS ontology models, which proposed QoS ontology frameworks aiming to formally describe arbitrary QoS parameters. From their on-going work, we are aware that they did yet consider QoS-based service selection. Additionally, in [12], WSDL (Web Services Description Language) is extended to express QoS directly on service interfaces. WSDL is designed to encode functionality. Typically, functional aspects are more fixed than non-functional aspects. Flexibility is achieved by using separate files to encode QoS. QoS attributes can change without changing WSDL files.

There are projects studying QoS-empowered service selection. In [18], authors present a QoS-aware middleware-supporting quality-driven Web service composition. Two service selection approaches for constructing composite services have been proposed: local optimization and global planning. Their study proves that global planning is better than local optimization. Authors in [1] study a similar approach in service selection with QoS constraints in global view. Both service selection methods are based on integer linear programming and best suited for small-size problems as its complexity increases exponentially with the increasing problem size. In our solution, we adopted the global planning to enhance the coordinator’s performance and reliability so as to intelligently allocate composition tasks.

Most existing work targeting on P2P-based Web service selection includes several major relevant proposals. In [3], service selection is based on historic information of service execution. A separate registry is used to store this information and policies are used. METEOR-S [6] and HyperCup [11] base the distribution of semantic Web service descriptions on a classification system expressed in service or registry ontologies. In our opinion, these solutions are good in terms of globally organizing registries to benefit service management rather than for the service discovery or selection itself. Though it is relatively effective to publish and update service description information based on their categories, it would be difficult for service requestors to select certain services without understanding details of their principles. In contrast, our UOW-SWS is built by taking considerations of new intuitive correlations between various service quality measurements and also testified upon a well-founded peer-to-peer e-service workflow system, which the authors have developed in the past [16].

6. Conclusion and Future Work

In this paper, we discussed the significance of non-functional properties in WSMO for facilitating P2P-based service selection and task allocation. We not only augmented WSMO description by involving QoS perspectives, but also designed and implemented a generic algorithm to facilitate the peer selection.

With regard to our further work, we foresee that developing some efficient and close-optimal algorithms (e.g. Ant Colony Optimization and Particle Swarm Optimization) would greatly improve the whole system's computing performance for large-scale services composition and also fit better within real application cases. In addition, further research will be concerned with more complicated application of our approach to other domain-specific situation settings.

References

[1] R. Aggarwal, K. Verma, J. Miller and W. Millnor, "Constraint drivenWeb service composition in METEOR-S". In Proceedings of the IEEE International Conference on Service Computing (SCC'2004). Shanghai, China.

[2] S. Bleul, T. Weise, K. Geihs, "An Ontology for Quality-Aware Service Discovery", special issue on "Engineering Design and Composition of Service-Oriented Applications", Computer Systems Science Engineering, 2006.

[3] D. Chakraborty, S. K. Jaiswal, A. Misray, and A. Nanavati. Middleware architecture for evaluation and selection of 3rdparty Web services for service providers. In Proc. of the IEEE Int'l Conf. on Web Services, pages 647–654. IEEE CS, 2005.

[4] JXTA: JXTA v2.0 Protocols Specification. Available at: http://jxta-spec.dev.java.net /JXTAProtocols.pdf

[5] K. Lee, J. Jeon, W. Lee, S. Jeong and S. Park, "QoS for Web services: Requirements and Possible Approaches". W3C Working Group Note 25, 2003. Available at: http://www.w3c.or.kr/kr-office/TR/2003/ws-qos/.

[6] LSDIS: METEOR-S: Semantic Web services and Processes. Available at: http://lsdis.cs.uga.edu/projects/ meteor-s/

[7] D. A. Menascé, "QoS issues in web services", In IEEE Internet Computing, IEEE Press, pages 72-75, November-December 2002.

[8] I.V. Papaioannou, D.T. Tsesmetzis, I.G. Roussaki and E.A. Miltiades, "QoS Ontology Language for Web-Services". In proceedings of the 20th International Conference on Advanced Information Networking and Applications (AINA 2006), Volume 1, pp.18-20, IEEE Press.

[9] S. Ran, "A model for Web services Discovery with QoS". ACM SIGecom Exchanges, 4(1): 1-10.

[10] D. Roman, U. Keller, H. Lausen, et. al., "Web Service Modelling Ontology". Applied Ontology, 1(1): 77-106, 2005.

[11] M. Schlosser, M. Sintek, S. Decker and W. Nejdl, "A scalable and ontology-based P2P infrastructure for semantic Web services". In Proceedings of the Second International Conference on Peer-to-Peer Computing (P2P'2002), page 104.

[12] A. ShaikhAli, O. F. Rana, R. Al-Ali, and D. W. Walker. UDDIe: An extended registry for Web services. In Proc. of the Symp. on Applications and the Internet, pages 85–89. IEEE CS, 2003.

[13] J. Shen, A. Krishna, S. Yuan, K. Cai, Y.M. Qin, "A Pragmatic GIS-Oriented Ontology for Location Based Services", the 19th Australian Software Engineering Conference (ASWEC 2008), Perth, Australia, published by IEEE Computer Society Press, March, 2008, pp.562-569.

[14] I. Toma, D. Foxvog and M. C. Jaeger, "Modeling QoS characteristics in WSMO". Workshop on Middleware for Service Oriented Computing (MW4SOC'06), pp.42-47.

[15] D.T. Tsesmetzis, I.G. Roussaki, I.V. Papaioannou and M.E. Anagnostou: QoS awareness support in Web-Service semantics, Proceedings of the Advanced International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW 2006), p.128.

[16] J. Yan, Y. Yang and G. Raikundalia, "SwinDeW - a p2p based decentralized workflow management system". IEEE Transactions on Systems, Man and Cybernetics - part A, 36(5): 922 – 935, 2006.

[17] S. Yuan and J. Shen, "Mining E-Services in P2P-based Workflow Enactments", special issue "Web Mining Applications in E-commerce and E-services" of Online Information Review, Emerald Group Publishing. Vol. 32 (2), pp: 163-178. 2008.

[18] L. Zeng, B. Benatallah, A. Ngu, M. Dumas, J. Kalagnaanam and H. Chang, "QoS-aware middleware for Web services composition". IEEE Transaction on Software Engineering. Vol. 30 (5), pp: 311 - 327. May 2004.